ITL.1046US
(P17448)

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: ERROR CORRECTION FOR MULTI-LEVEL CELL MEMORY WITH OVERWRITE CAPABILITY

INVENTOR: RICHARD E. FACKENTHAL

Express Mail No. EV 337934018US
Date: October 16, 2003

Prepared by: Joseph C. Arrambide
Trop, Pruner & Hu, P.C.
8554 Katy Freeway, Ste. 100, Houston, TX 77024
713/468-8880 [Office], 713/468-8883 [Fax]

ERROR CORRECTION FOR MULTI-LEVEL CELL MEMORY WITH OVERWRITE CAPABILITY

Background

This invention relates to semiconductor memories and particularly to multi-level cell (MLC) flash memories with error correction code.

A flash memory cell, as well as other types of memory cells, may be configurable to store more than one voltage threshold level (V_T) in one cell. Depending on the number of threshold levels, more than one bit may be stored per cell. For example, a memory cell capable of storing four threshold levels is capable of storing two bits per cell.

Aggressive scaling of process technology and demand for higher density chips present increasing reliability challenges to multi-level cell memory product development. Error correction codes (ECC) are used throughout the electronics, communications, and memory industries to counter low-level reliability problems while improving performance or reducing cost per bit. A typical error correction method involves calculating parity bits for a codeword of data. For example, a 256 bit codeword may have parity bits associated with it. The parity bits are calculated based on an error correction code.

A fundamental problem with designing error correction codes into a MLC flash memory is the one-way nature of flash programming. Error correction codes work on codewords (or other data segments). Parity bits (or parity check bits) may be generated and then associated with a codeword. Each time that a codeword is rewritten, the associated parity bits will have to change.

However, the parity bit may only change from a one to a zero because rewriting a bit from zero to one in flash memory involves a block erase. In other words, error correction codes are not practical with flash memories, which may be rewritten several times, since the associated parity bits may need to change from zero to one in response to the rewriting of a codeword.

Therefore, there is a need for alternative ways to implement error correction methods in flash memories, while allowing overwriting.

Brief Description of the Drawings

Figure 1 is a block diagram of a system which may have one embodiment of the present invention.

Figure 2 is a table of logic states and voltage thresholds of one embodiment of the present invention.

Figure 3 depicts bit positions in memory cells of one embodiment of the invention.

Figure 4 is a depiction of a memory map in accordance with one embodiment of the present invention.

Figure 5 is a flowchart implementing aspects in accordance with one embodiment of the present invention.

Detailed Description

Referring to Figure 1, an electronic system 100 may include a processor 110 coupled to a system bus 120. A multi-level cell memory 140 may be coupled through the bus 120 to the processor 110. Similarly, a wireless interface 130 may be coupled to the processor 110 through bus 120. The multi-level cell memory 140 may include a controller 150 and a memory array 160. The controller 150, in one embodiment, may store a write algorithm 152.

In one embodiment, the system 100 may enable wireless network access using a wireless interface 130. The wireless interface 130 may be a radio frequency interface, as one example, including a transceiver and an antenna. The antenna may be a dipole antenna, or a helical antenna as two examples. For example, the wireless interface 130 may enable access to a server or a client on a client server system over a suitable wireless medium, such as a radio frequency medium. However, the present invention is not limited to a processor-based system that permits wireless access.

A multi-level cell memory may operate at a density of bits per cell that is less than the density capacity of the cell. Thus, one cell may be at a higher density than another cell in the same array, even though both cells have

the same capacity to store a given number of bits greater than one bit.

In the example of Figure 2, four voltage threshold levels L0, L1, L2, and L3 are illustrated. L0 corresponds to a data bit pattern of 11 and L3 corresponds to the bit pattern 00. In one example, a higher density mode may utilize all four levels, thus storing two bits of data per multi-level cell. In one embodiment, a lower density mode may utilize just two voltage threshold levels such as the levels L0 and L3 and, in such case, only one bit of data is stored in the lower density mode.

In other examples, higher density modes may include utilizing any number of voltage threshold levels up to and including the maximum number of levels that the cell may handle. Similarly, a lower density mode may include using any smaller number of threshold voltage levels, when compared to a higher density mode.

In the example of Figure 2, the high bit is the more significant bit and the low bit is the less significant bit. If only one bit is stored, using levels L0 and L3, the low bit could effectively be ignored.

In multi-level cell codewords, bits stored in the same cell may be split apart so that they are not located next to each other in the codeword. In one embodiment, more significant bits of a codeword may be segregated from less significant bits. For example, the bits of a two bit

multi-level cell may be split such that the more significant bits of the multi-level cells are located on a first side of a codeword while the less significant bits are located on the opposite side of the codeword. In effect, more significant bits may be grouped together within the codeword even though those bits may be from different cells. The segregation of more significant bits within a codeword may ultimately facilitate more ready access to the more significant bits when the codeword is in a lower density mode.

A bit arrangement for codeword 400a is shown on the left in Figure 3. A codeword 400 has n bits. In the codeword 400a, the two bits of the multi-level cells may be logically adjacent in the codeword. In this example, bits 0 (more significant bit) and 1 (the less significant bit) of "one cell" may be stored side-by-side in the codeword 400a.

In another bit arrangement, codeword 400b on the right in Figure 3, bits from the same cell are split apart within the codeword. More significant bits from cells making up the codeword 400b may be in section A of the codeword 410 and the less significant bits of the same cells may be in the section B of a codeword 400b. In one embodiment, the more significant bits are segregated in the first half of the codeword from the less significant bits in the second half of the codeword.

In this embodiment, the controller 150 may see n/2 valid bits in the first half of the codeword and n/2 less significant bits in the second half of the codeword. Flash file systems (which may include low level media drivers) then may advantageously ignore the section B of the codeword 400b, in this example.

In general, more significant bits from a number of cells may be packed next to one another in a first codeword section. Then less significant bits may be placed in a different or second codeword section. The order, arrangement, and number of such codeword sections is open to wide variations.

Part of the array 160 (Figure 1) may include a memory space 500 storing eight codewords A-H as shown in Figure 4. A density flag 510 may be associated with each codeword in one embodiment. If more than two bit densities per cell are utilized, the density flag may use two or more bits in this embodiment. In one embodiment, when the density flag 510 is "1," the codeword is in a higher density mode (such as two bits per cell). Conversely, when the write density flag 510 is "0," the codeword is in a lower density mode (such as one bit per cell). Of course, other coding protocols may be utilized to indicate the selected bit density. When a codeword is in a lower density mode, the codeword (such as the codeword C and E), may have invalid

areas (illustrated in Figure 4 by the slashed areas) for the less significant bits, in this example.

In one embodiment, using at least one higher and at least one lower density mode, when a block of memory is erased, all codewords in the block default to the higher density mode and all write density flags are reset to indicate the higher density mode. Those erased cells are then available to be written to a selected density mode such as a lower density mode.

In a lower density mode, the less significant bits 530 can be filled with either the same data as the more significant bits 520, or all zeros, or all ones. Filling the less significant bits 330 with the same data as the more significant bits 320 may disguise the lower density mode from the internal multi-level cell programming algorithm.

When the system reads back a lower density mode codeword, the high bit data 520 may be received. Since the more significant bits 320 are reliable to an L0 or L3 voltage threshold in this example, error correction code (ECC) may not be used. Since error correction code (ECC) is not executed, the system 100 may write the more significant bits of the lower density mode codewords many times (writing only from a "1" to a "0").

Parity bits $\mathbf{P}^i$, $0 \leq i \leq m$, may be included in the column 540 of the codewords A, B, E, F, G and H in the

higher density mode while no parity bits are used in the codewords C and E that are in the lower density mode in one embodiment. Since the lower density mode codewords, such as the codewords C and E in one example, do not have parity bits associated with them, they may be overwritten in this embodiment.

Referring to Figure 5, the write algorithm 152 may be implemented in software, firmware, microcode or hardware. Initially, the write algorithm 152, that may be executed by the controller 150, determines if a particular codeword may need to be rewritten, as shown block 220. In one embodiment, if overwriting is needed, then the memory 140 sets a "lower density" flag as shown in block 250. The memory 140 may then complete the re-write by writing to memory in a lower density mode as shown in block 260.

In cases where overwriting may be necessary, it is advantageous to use the lower density mode thereby eliminating the need for error correction codes. This is because in the lower density mode, the likelihood of an error is less and therefore it is more feasible to dispense with the error correcting code.

With the error correcting code, it is generally not feasible to overwrite the data in flash memory. Thus, to enable overwriting, a lower density may be selected. Selecting the lower density improves the reliability of the stored data reducing the need for error correcting code.

During a write, the memory 140 may then read a density flag to see if the subject codeword has been flagged as a lower density codeword as shown in diamond 230. In one embodiment, if the codeword has been flagged as a lower density codeword, then the memory 140 writes to the array 160 in a lower density mode, as indicated in block 260. If the codeword has not been flagged as a lower density codeword, then the memory 140 may generate a parity bit (or other ECC) and writes to memory in a higher density mode, as indicated in block 240. In one embodiment, this would complete the write as indicated in block 270.

An embodiment is illustrated in which software control is utilized. Of course, hardwired embodiments may be utilized as well. In addition, while embodiments of the present invention are described in connection in which only one write mode status bit are utilized, the present invention is applicable to any number of write mode status bits greater than one and multi-level memory cells with any number of levels.

While the present invention has been described with respect to a limited number of embodiments, those skilled in the art will appreciate numerous modifications and variations therefrom. It is intended that the appended claims cover all such modifications and variations as fall within the true spirit and scope of this present invention.